Assignment-2

Name of the Student (Team Lead)	M.Nandhakumar
Register Number	820519205024
Maximum Mark	2 Mark

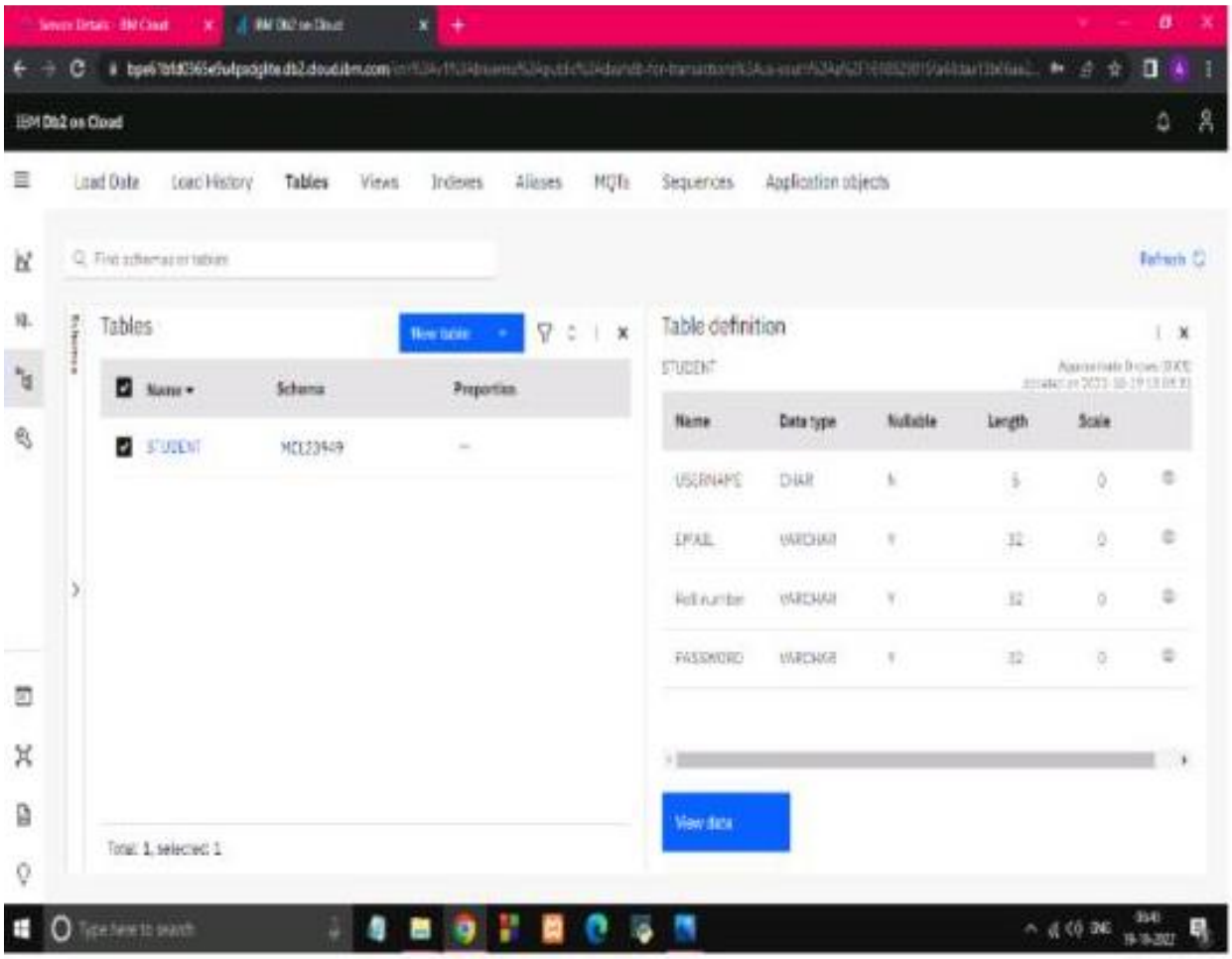
Questions:

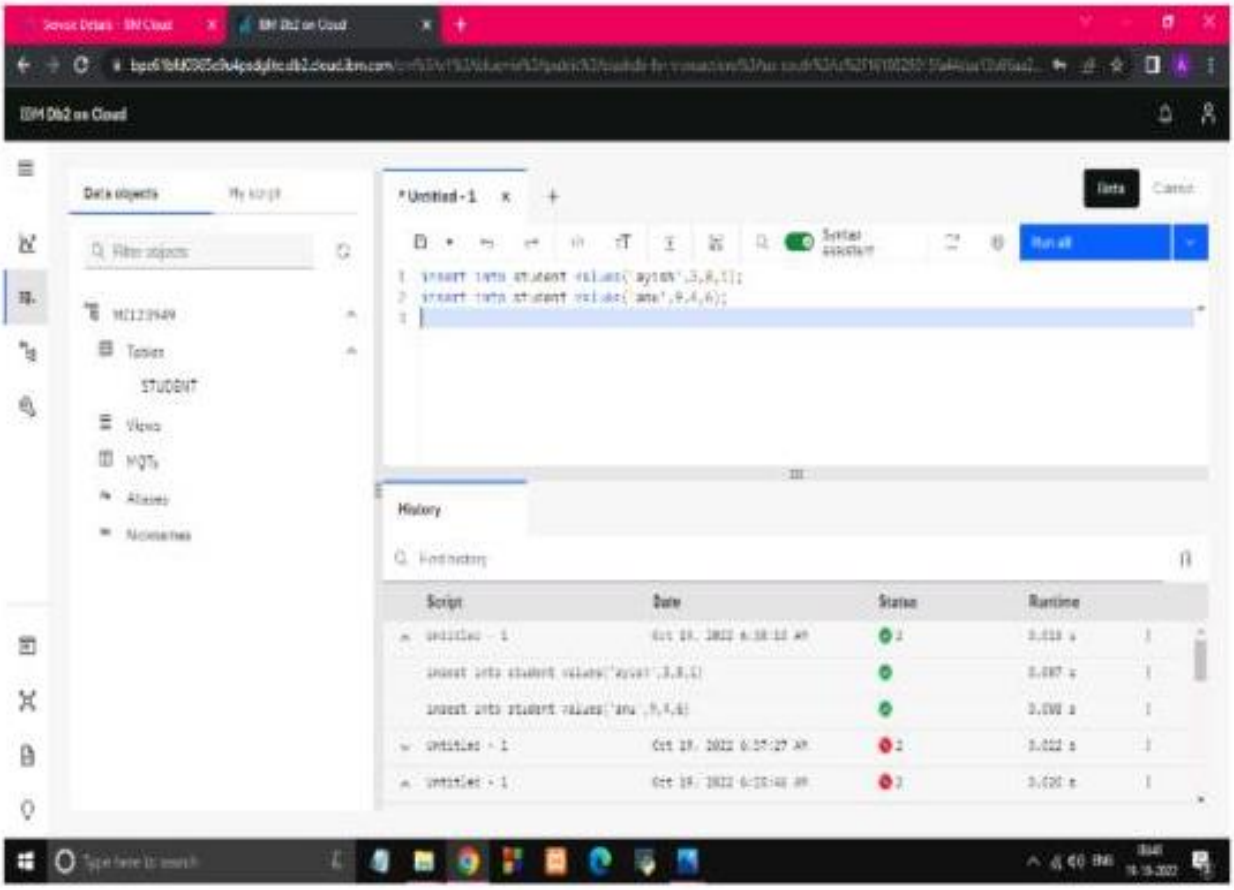
Create User table with user email, username, roll number, password.

1. Perform UPDATE, DELETE Queries with user table.
2. Connect python code to db2.
3. Create a flask app with registration page, login page and welcome page. By default load the registration page once the user enters all the fields store the data in database and navigate to login page authenticate user username and password. If the user is valid show the welcome page.

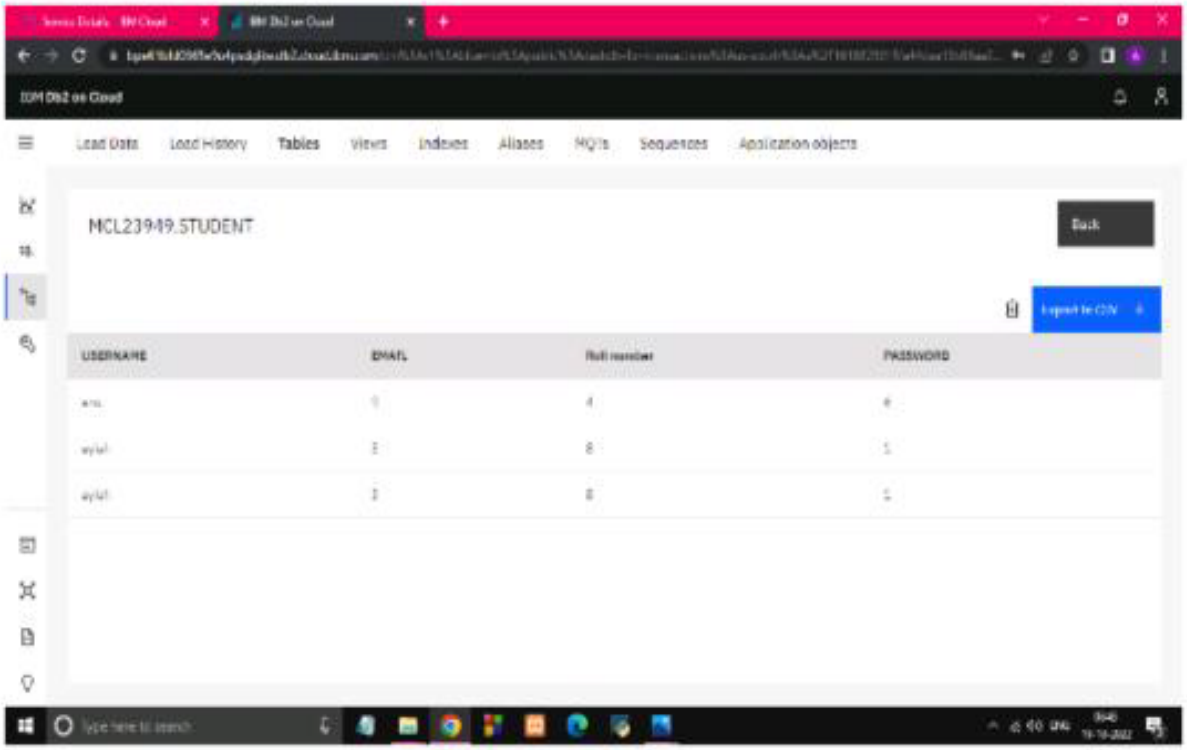
Solutions:

1. Creating user table with user email, username, roll number, password.



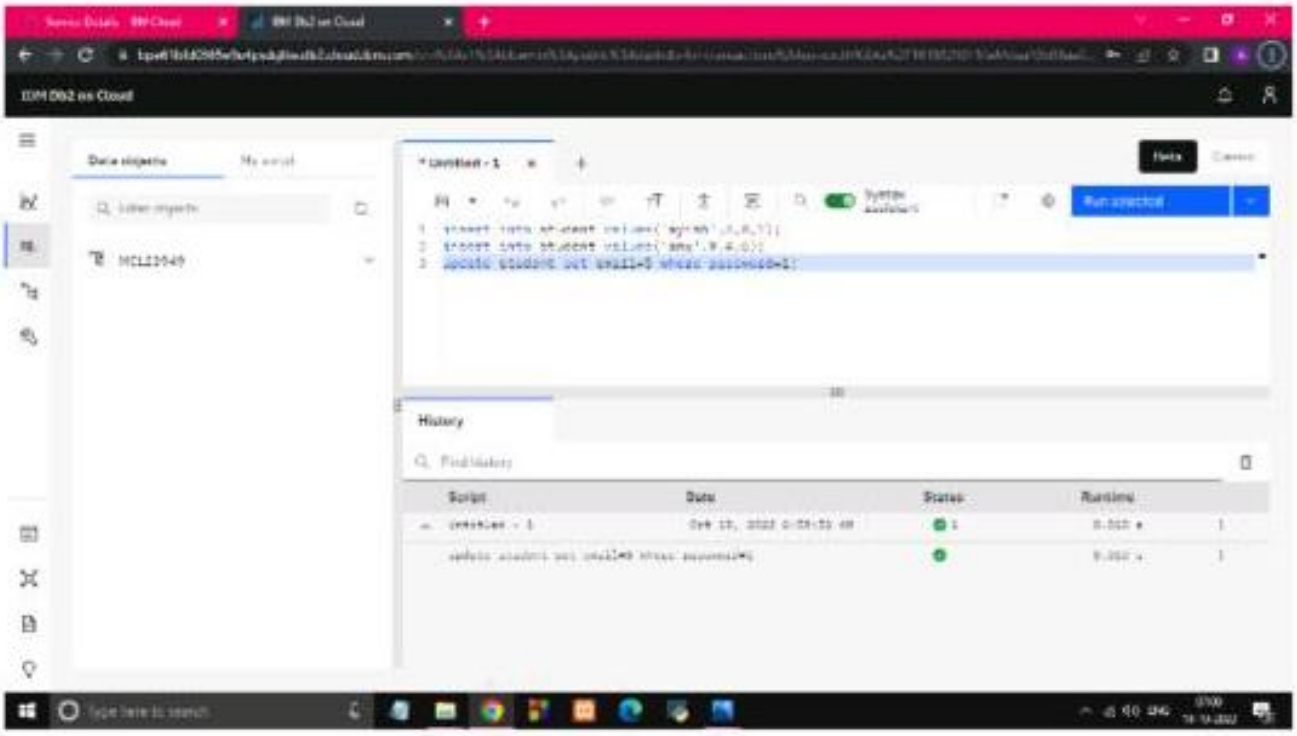

IBM Db2 on Cloud
Data objects
My script
Filter objects
STUDENT
Tables
Views
MQTs
Aliases
Nicknames
* Untitled - 1
Data
Cancel
Run all
insert into student values('ayish',3,8,1);
insert into student values('anu',9,4,6);
History
Script
Date
Status
Runtime

Output:

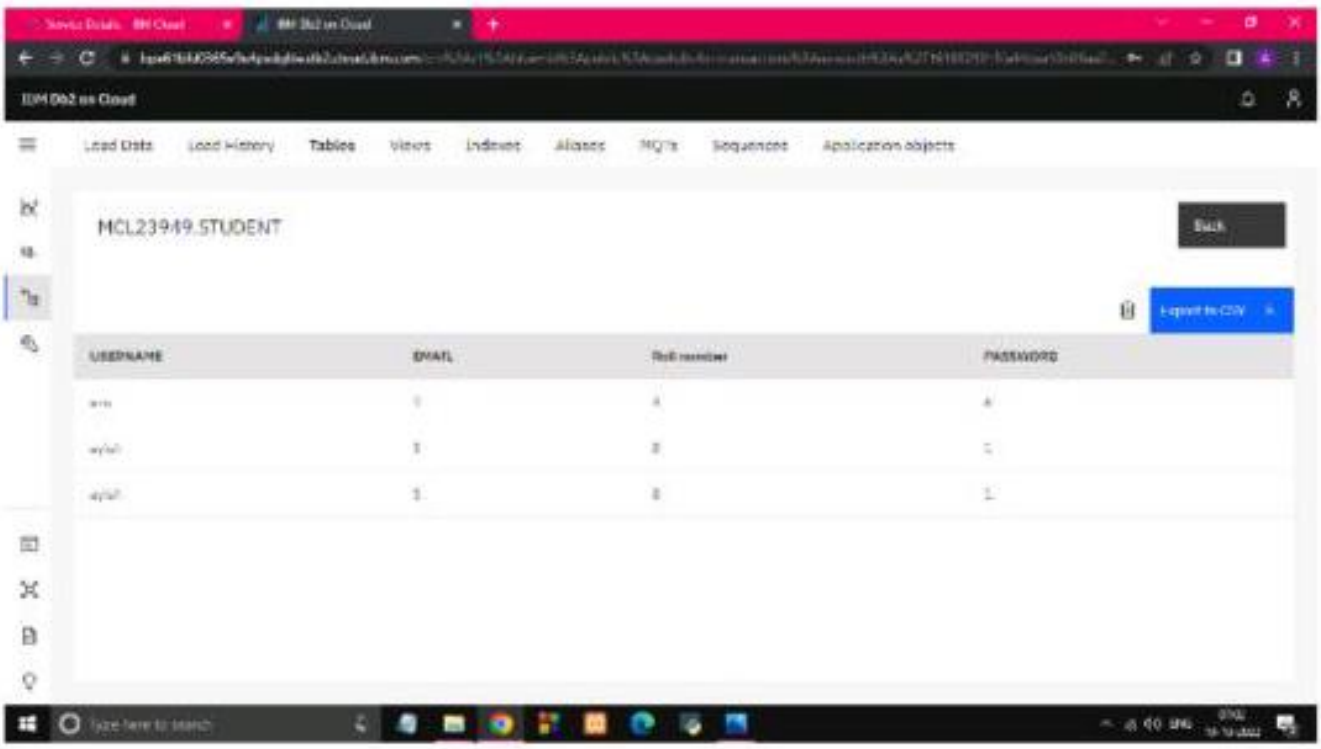

IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
MCL23949.STUDENT
Back
USERNAME
EMAIL
PASSWORD

2. Performing UPDATE, DELETE Queries with user table

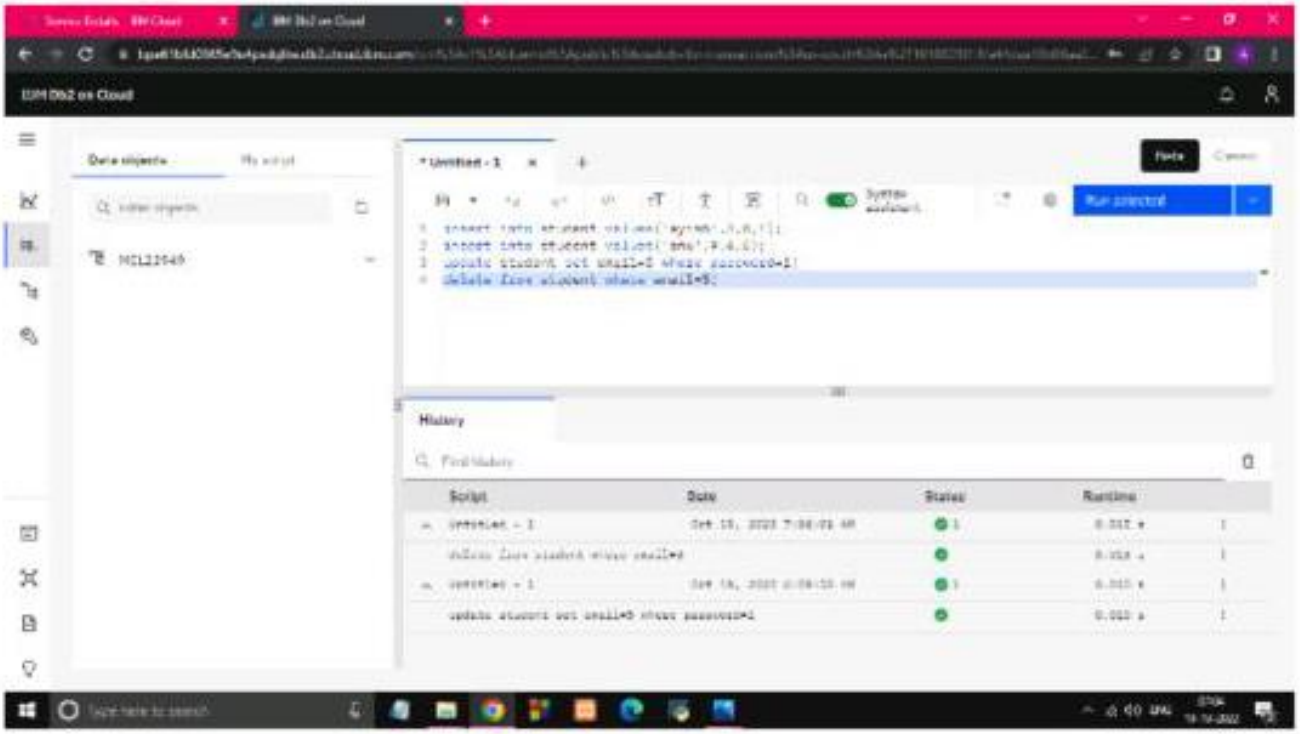
UPDATE:



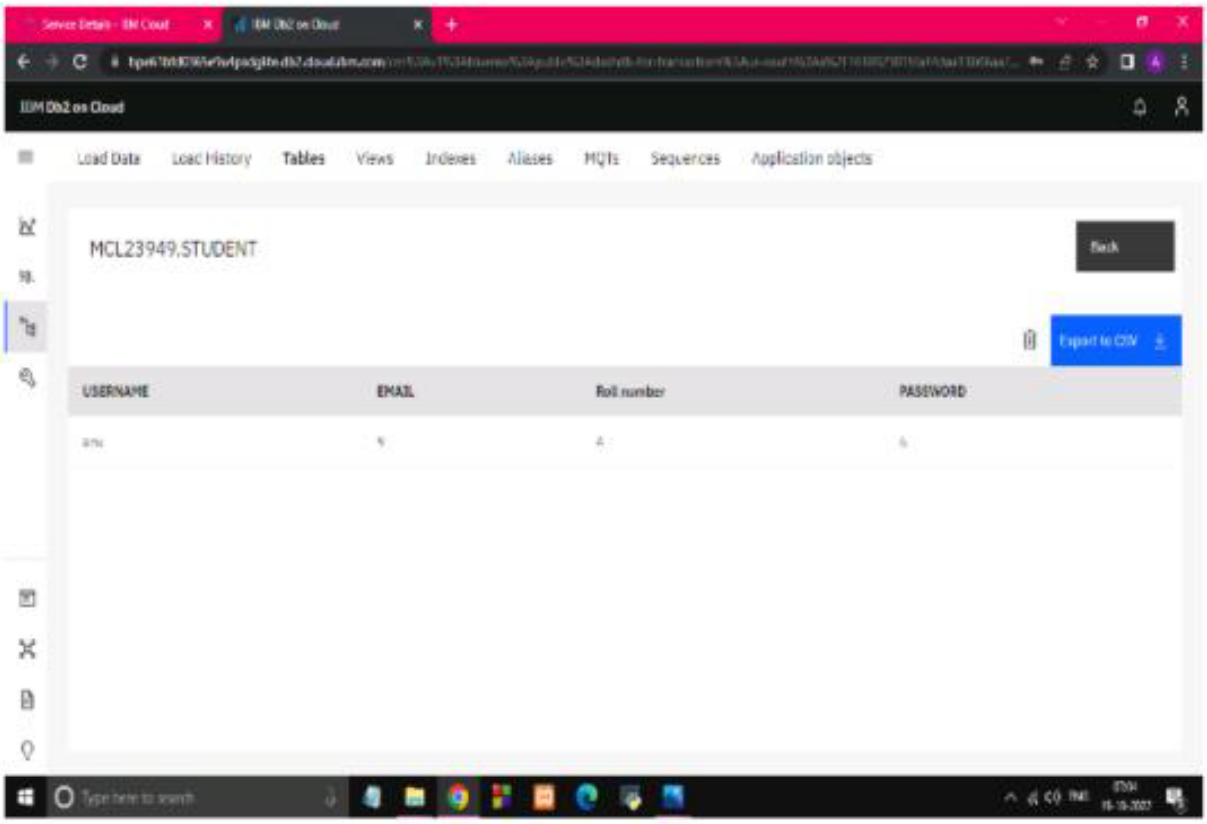
OUTPUT:


Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
MCL23949.STUDENT
USERNAME
EMAIL
PASSWORD

DELETE:



OUTPUT:


IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
MCL23949.STUDENT
Back
Export to CSV
USERNAME
EMAIL
Roll number
PASSWORD

3.Connect python code to db2.

```
import  ibm_db

conn=

ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf9-0fbb
7e483086.clogj3sd0tgtu0lqde00.datbases.appdomain.cloud;PORt=32459;SECURI
TY=SSl;SSLServerCertificate=DigiCertGlobalRoot.crt;UID=mcl23949;PWD=dmUiv
2o6zjDMw0Ea",)

print(conn)

print("connection successful...")
```

4.Creating a flask app with registration page, login page and welcome page.

```
# Store this code in 'app.py' file
```

```
from flask import Flask, render_template, request, redirect, url_for, session

import ibm_db

import re

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=9938aec0-8105-433e-8bf
9-0fbb7e483086.clogj3sd0tgtu0lqde00.datbases.appdomain.cloud;PORt=32459;
SECURITY=SSl;SSLServerCertificate=DigiCertGlobalRoot.crt;UID=mcl23949;PWD=
dmUiv2o6zjDMw0Ea",)


app = Flask(__name__)

app.secret_key='a'

conn = ibm_db.connect()

@app.route('/')

@app.route('/login', methods =['GET', 'POST'])

def login():

msg = ''

if request.method == 'POST' and 'username' in request.form and 'password' in
request.form:

username = request.form['username']

password = request.form['password']

sql=('SELECT * FROM users WHERE username = % s AND password = % s',
(username, password, ))

stmt = ibm_db.prepare(conn,sql)

ibm_db.bind_param(stmt,1,username)

ibm_db.bind_param(stmt,2,password)

ibm_db.execute(stmt)

account = ibm_db.fet_assoc(stmt)
```

```
print(account)
if account:
session['loggedin'] = True
session['id'] = account['id']
session['username'] = account['username']
msg = 'Logged in successfully !'
return render_template('index.html', msg = msg)
else:
msg = 'Incorrect username / password !'
return render_template('login.html', msg = msg)


@app.route('/logout')
def logout():
session.pop('loggedin', None)
session.pop('id', None)
session.pop('username', None)
return redirect(url_for('login'))


@app.route('/register', methods =['GET', 'POST'])
def register():
msg = ''
if request.method == 'POST' and 'username' in request.form and 'password' in
request.form and 'email' in request.form :
username = request.form['username']
password = request.form['password']
```

```
email = request.form['email']

sql=('SELECT * FROM users WHERE username = % s AND password = % s',
(username, password, ))

stmt = ibm_db.prepare(conn,sql)

ibm_db.bind_param(stmt,1,username)

        ibm_db.execute(stmt)

        account = ibm_db.fet_assoc(stmt)

        print(account)

if account:

msg = 'Account already exists !'

elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):

msg = 'Invalid email address !'

elif not re.match(r'[A-Za-z0-9]+', username):

msg = 'Username must contain only characters and numbers !'

elif not username or not password or not email:

msg = 'Please fill out the form !'

else:

insert_sql=('INSERT INTO accounts VALUES (NULL, % s, % s, % s)', (username,
password, email, ))

prep_stmt = ibm_db.prepare(conn,insert_sql)

ibm_db.bind_param(prep_stmt,1,username)

        ibm_db.bind_param(stmt,2,password)

        ibm_db.bind_param(stmt,3,email)

        ibm_db.execute(prep_stmt)

msg = 'You have successfully registered !'

elif request.method == 'POST':
```

```
msg = 'Please fill out the form !'
return render_template('register.html', msg = msg)
  if __name__=='__main__':
    app.run(host='0.0.0.0')
```

register.html:

```
<html>
<head>
<meta charset="UTF-8">
<title> Register </title>
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body style="background-color:powderblue;"></br></br></br></br></br>
<div align="center">
<div align="center" class="border">
<div class="header">
<h1 class="word">Register</h1>
</div></br></br></br>
<h2 class="word">
<form action="{{ url_for('register') }}" method="post">
<input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/></br></br>
<input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox"/></br></br>
<input id="email" name="email" type="text" placeholder="Enter Your Email ID" class="textbox"/></br></br>
```

```
<input type="submit" class="btn" value="Sign Up"></br>

</form>

</h2>

<p class="bottom">Already have an account? <a class="bottom"
href="{{url_for('login')}}"> Sign In here</a></p>

</div>

</div>

</body>

</html>
```

login.html:

```
<html>

<head>

<meta charset="UTF-8">

<title> Login </title>

<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

</head>

<body></br></br></br></br></br>

<div align="center">

<div align="center" class="border">

<div class="header">

<h1 class="word">Login</h1>

</div></br></br></br>

<h2 class="word">

<form action="{{ url_for('login') }}" method="post">
```

```
<input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/></br></br>

<input id="password" name="password" type="password" placeholder="Enter Your Password" class="textbox"/></br></br></br>

<input type="submit" class="btn" value="Sign In"></br></br>

</form>

</h2>

<p class="bottom">Don't have an account? <a class="bottom" href="{{url_for('register')}}"> Sign Up here</a></p>

</div>

</div>

</body>

</html>
```

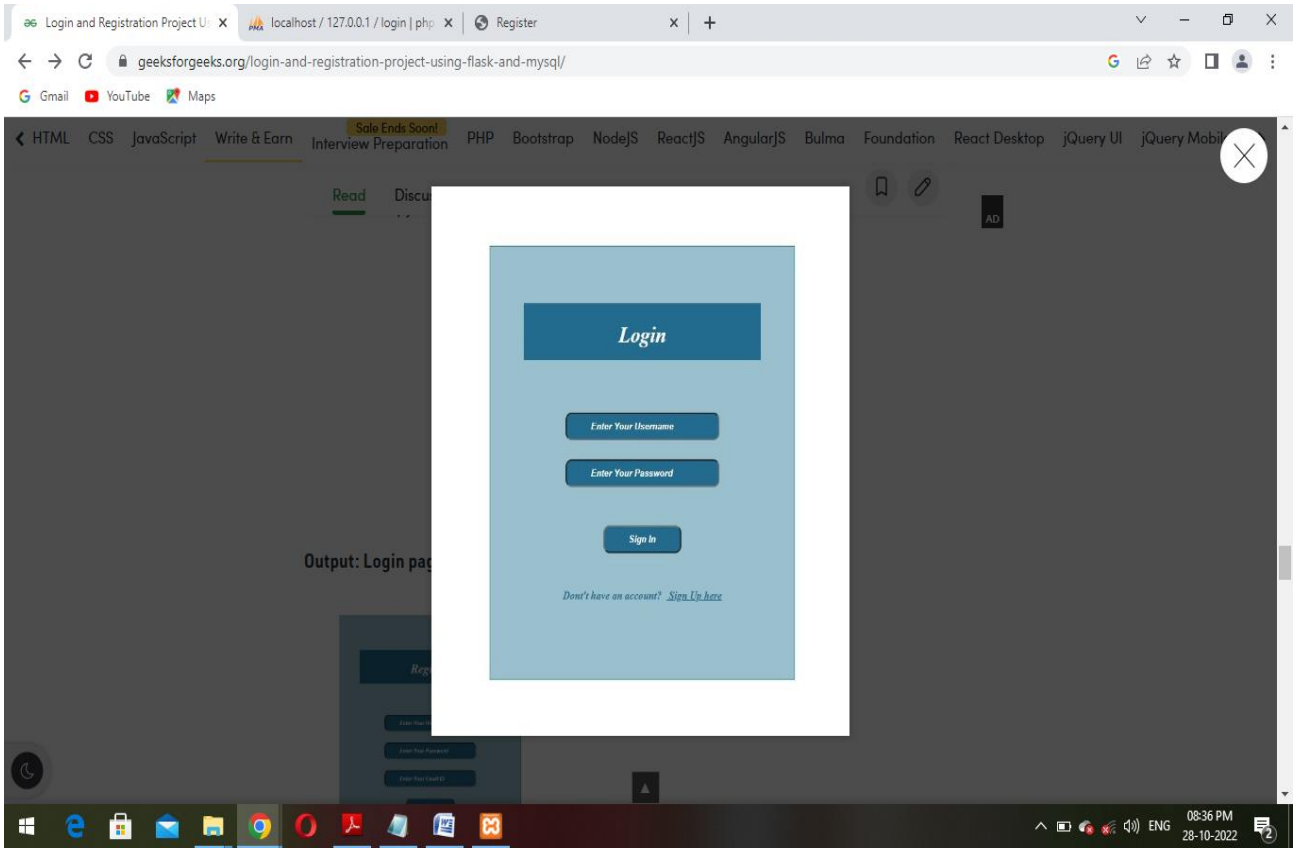
Index.html:

```
<html>

<head>

<meta charset="UTF-8">

<title> Index </title>

<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

</head>

<body></br></br></br></br></br>

<div align="center">

<div align="center" class="border">

<div class="header">

<h1 class="word">Index</h1>
```

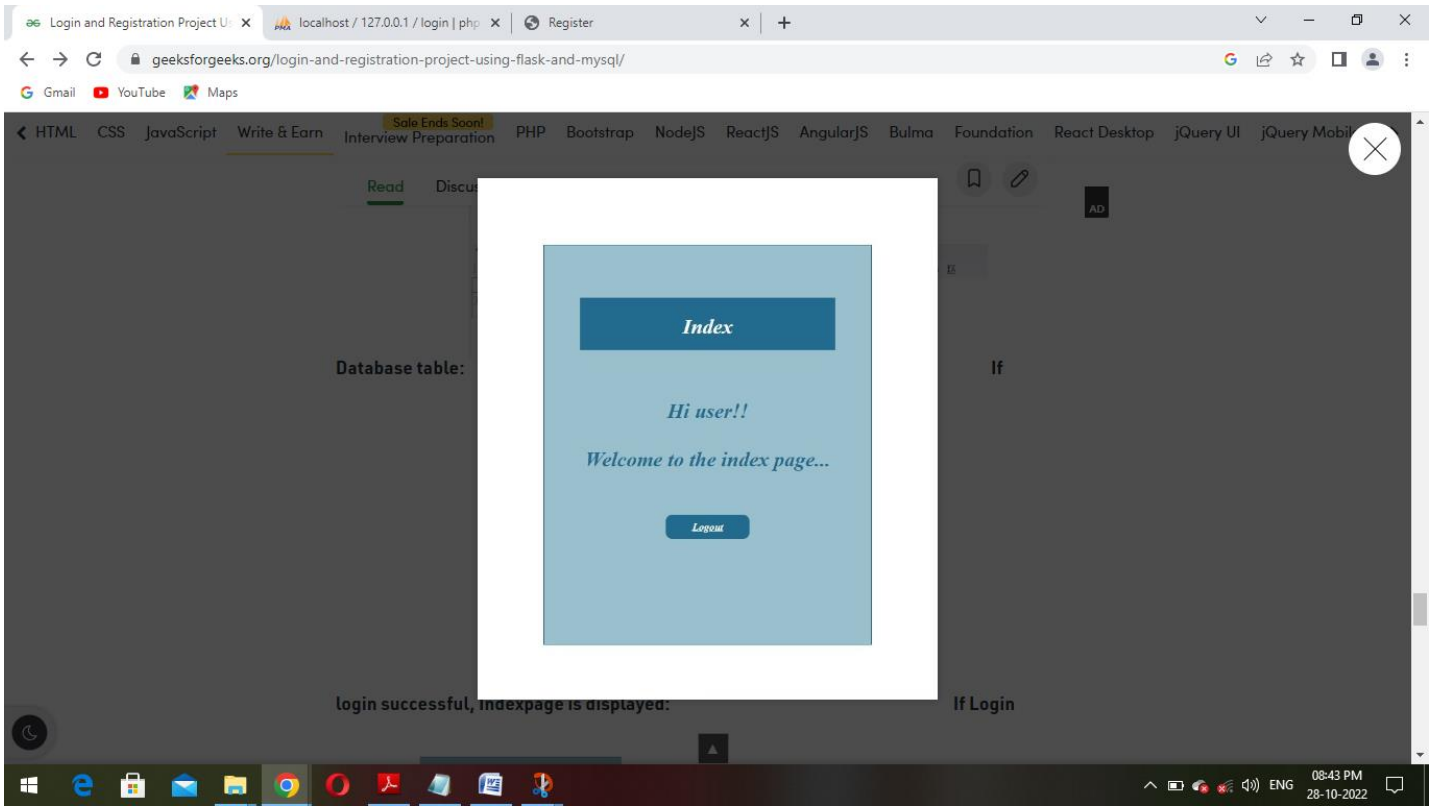
```
</div></br></br></br>
<h1 class="bottom">
Hi {{session.username}}!!</br></br> Welcome to the index page...
</h1></br></br></br>
<a href="{{ url_for('logout') }}" class="btn">Logout</a>
</div>
</div>
</body>
</html>
```

OUTPUT

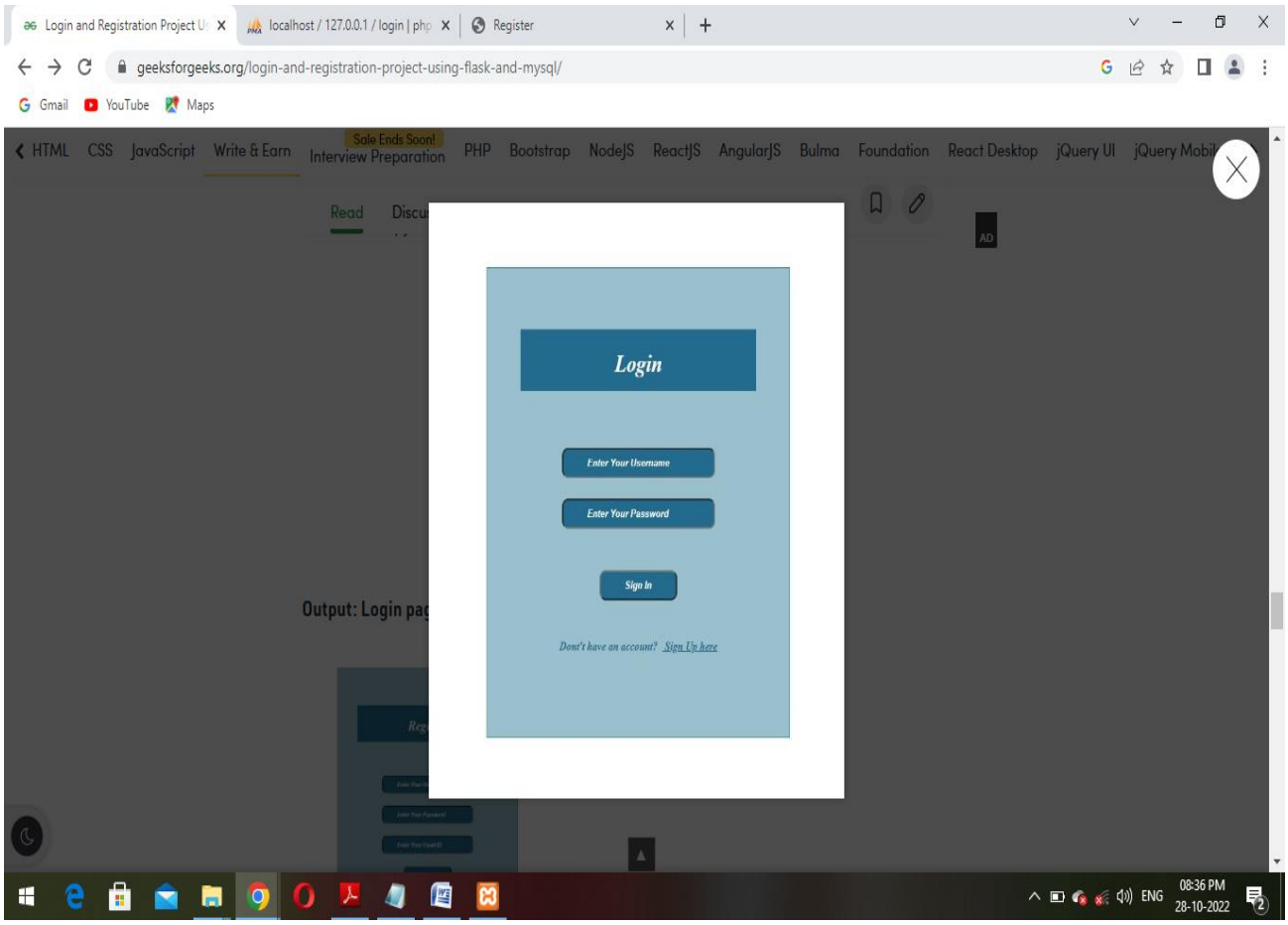
Register page:



Welcome Page:



Login Page:


Login and Registration Project U
localhost / 127.0.0.1 / login | php
Register
geeksforgeeks.org/login-and-registration-project-using-flask-and-mysql/
Gmail
YouTube
Maps
HTML
CSS
JavaScript
Write & Earn
Sale Ends Soon!
Interview Preparation
PHP
Bootstrap
NodeJS
ReactJS
AngularJS
Bulma
Foundation
React Desktop
jQuery UI
Read
Login
Enter Your Username
Enter Your Password
Sign In
Don't have an account? Sign Up here
Output: Login pag
ENG
08:36 PM
28-10-2022